

Introducing DHCP

Dru Lavigne

In the next few articles, I'll be covering the DHCP protocol: how it works and how to configure a DHCP client and a DHCP server using FreeBSD. If you've ever connected a computer to a network, you've probably heard of DHCP, the Dynamic Host Configuration Protocol. As its name suggests, this protocol is designed to configure a host dynamically with the TCP/IP information it needs in order to communicate on a network.

The alternative to dynamic addressing is static addressing. Static addressing occurs when you manually type in an IP address, subnet mask, and default gateway address. FreeBSD systems support both static and dynamic addressing. Typically, you would use static addressing to set up a small home network and dynamic addressing when you connect to the Internet.

Knowing Your Interfaces

Before using any type of addressing, you need to determine the FreeBSD name for your network interface card. To do that, use the `ifconfig` command like so:

```
$ ifconfig -a
rl0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> mtu 1500
    ether 00:05:5d:d2:19:b7
    media: Ethernet autoselect
ed0: flags=8843<UP,BROADCAST,SIMPLEX,MULTICAST> mtu 1500
    ether 00:50:ba:de:36:33
    media: Ethernet autoselect
lp0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    inet 127.0.0.1 netmask 0xff000000
```

The `ifconfig` command will display both physical and virtual interfaces. You can recognize the Ethernet interfaces as they will have an "ether" or MAC address. The FreeBSD name for the adapter will have 2 or 3 letters followed by a number. In this example, there are two Ethernet NICs; one is named `rl0` and the other `ed0`. There are also two virtual interfaces, `lp0` and `lo0`. You may recognize 127.0.0.1 as the "loopback address", meaning `lo0` is the name of the loopback virtual interface.

Most interfaces have an entry in section 4 of the manual. In my case, I could try `man 4 rl`, `man 4 ed`, `man 4 lp`, and `man 4 lo` for more information about each interface. Note that you don't include the number when specifying the interface name with `man`. Instead, the interface number indicates how many interfaces of that type are installed; the count starts at 0. For example, if I had had two realtek NICs, they would be called `rl0` and `rl1`. (That's "arr-ell", not "arr-one".)

Static Addresses

To assign a static IP address and subnet mask to the `rl0` interface, I'll become the superuser and use `ifconfig` like so:

```
# ifconfig rl0 192.168.10.1 255.255.255.0
```

When you try this command, don't forget to specify the correct interface name for your system or you will receive an error message. To see if your command was successful, ask `ifconfig` to limit its information to one particular interface, rather than using the `a` switch to see all the interfaces:

Introducing DHCP

Dru Lavigne

```
# ifconfig r10
```

```
r10: flags=8802<BROADCAST,SIMPLEX,MULTICAST> mtu 1500
    inet 192.168.10.1 netmask 0xfffff00 broadcast 255.255.255.0
    ether 00:05:5d:d2:19:b7
```

If your computer also requires a default gateway address, use the route command like so:

```
# route add default 192.168.10.25
```

To confirm your change:

```
# netstat -rn |grep G
Destination      Gateway          Flags           Refs           Use  Netif  Expire
default          192.168.10.25   UGSc           5              0    r10
192.168.10.1    127.0.0.1       UGHS           0              2    lo0
```

You'll note that I used grep to search the routing table for the G flag, which represents the gateway flag. The other flag to notice is U, which indicates that the gateway is up. This is always a good thing in a gateway.

This host is now configured, but none of those configurations will survive a reboot. It's inconvenient to retype in your static configurations every time you reboot a computer. Fortunately, you can tell your FreeBSD system to keep these configurations by including them in the system startup configuration file. On this host, I'll add the following lines to /etc/rc.conf:

```
ifconfig_r10="inet 192.168.10.1 netmask 255.255.255.0"
defaultrouter="192.168.10.25"
```

Be careful when making changes to this file, including getting the required quotation marks right. To test your change, make sure this command does not result in any errors:

```
# /etc/netstart
```

Dynamic Addresses

It's not a big deal to edit a few files if you only have one computer or a small network with just a few computers. However, if you are an administrator of a larger network, it is more convenient to use DHCP for two reasons. First, as the number of files you edit increases, so does the chance of typos and the possibility of two computers mistakenly being assigned the same IP address. Second, the more computers you have, the more inconvenient it is to have to sit down at each to enter their IP addressing information manually.

If your system is attached to a network that uses DHCP--the Internet, for example--you can take advantage of FreeBSD's built-in DHCP client. In my example, my second NIC (ed0) is attached to my cable modem. To receive an IP address from my ISP's DHCP server, I can use this command:

```
# dhclient ed0
```

Introducing DHCP

Dru Lavigne

To see if it worked:

```
# ifconfig ed0
```

```
ed0: flags=8843<UP,BROADCAST,SIMPLEX,MULTICAST> mtu 1500
      inet 2.2.2.2 netmask 0xffffffff broadcast 255.255.255.255
      ether 00:50:ba:de:36:33
```

To configure this system to use always dynamic addressing, I can add the following line to /etc/rc.conf:

```
ifconfig_ed0="DHCP"
```

DHCP Terminology

You need to know some DHCP terminology if you're going to be able to troubleshoot dhclient or to create your own DHCP server to use on your network.

When a DHCP client receives configuration information from a DHCP server, it is in the form of a "lease". This means that the client configuration remains valid for a limited period of time, configured on the DHCP server. The information contained in that lease was also configured on the server. A DHCP server can give out much more than just a host's IP address, subnet mask, and default gateway. It can inform the client of the IP addresses for the following types of servers in a network:

- cookie servers
- DNS servers
- finger servers
- font servers
- impress servers
- IRC servers
- log servers
- lpr servers
- NDS servers
- WINS servers
- NIS servers
- NIS+ servers
- NNTP servers
- NTP servers
- POP servers
- resource location servers
- SMTP servers
- streettalks servers
- swap servers
- TFTP server name
- time servers
- uap servers
- www servers

Introducing DHCP

Dru Lavigne

Whew, that's a lot of servers. In addition, DHCP can also set the client's MTU, TTL, hostname, and a few dozen other parameters. To see the complete list, check out `man dhcp-options`.

Once a DHCP client has a lease, it stores it in `/var/db/dhclient.leases`. Here is an example lease file:

```
lease {
  interface "ed0";
  fixed-address 2.2.2.2;
  option subnet-mask 255.255.240.0;
  option routers 2.2.2.1;
  option domain-name-servers 2.2.2.94,2.2.2.93,2.2.2.46;
  option broadcast-address 255.255.255.255;
  option dhcp-server-identifier 2.2.2.21
  option host-name "thishost";
  option domain-name "thisdomain.com";
  renew 3 2003/4/2 00:22:38;
  rebind 6 2003/4/5 02:50:06;
  expire 6 2003/4/5 23:50:06;
}
```

If you've been running `dhclient` for a while, you'll note that your lease file will have many such sections which start with `lease {` and end with the closing curly brace `}`. The information between the curly braces is the lease itself. Since leases don't last forever, each new lease assigned to the DHCP client is added to the end of the file. Your current lease will be at the very end of the file. If the file itself gets too long, it is copied over to `/var/db/dhclient.leases~` and a new `/var/db/dhclient.leases` file is started.

How does this whole lease process work? Let's assume for a moment that somewhere on your network is a properly configured DHCP server. This server will listen on UDP port 67, waiting for lease requests from DHCP clients. Let's also assume you have a brand new DHCP client with an empty `/var/db/dhclient.leases` file. Remember, you made that host a DHCP client by adding this line to `/etc/rc.conf`:

```
ifconfig_ed0="DHCP"
```

Nothing in that line indicates the IP address of the DHCP server. This is just as well because at this moment `ed0` doesn't even have an IP address; if it did, it wouldn't need to contact a DHCP server.

In order to contact the DHCP server, this host will send out a special packet known as a DHCPDISCOVER message. Since the host doesn't know the IP address of its DHCP server, it sends the packet to the broadcast address, `255.255.255.255`, in the hopes that a DHCP server will see it and respond. If you're familiar with networking, you know that broadcast packets are processed by all hosts that see the packet. However, only a DHCP server will understand the message in the packet as it is destined for UDP port 67.

You might also be aware that routers drop broadcast packets. This has a very big ramification for DHCP: if there isn't a DHCP server on your subnet or cable segment, the DHCP server on another subnet will never receive that broadcast. Does this mean that you need to have a DHCP server on

Introducing DHCP

Dru Lavigne

every subnet in your network? Fortunately, no. Instead, you use something known as a "bootp relay agent" to deliver that message to a DHCP server. I'll talk more about these relay agents in the next few articles.

Ideally a DHCP server will receive the DHCPDISCOVER message. When it does, it will check its database of available leases, setting one aside pending confirmation from the host. It will then send out a DHCPOFFER message containing the details of the lease. This message is sent to the DHCP client port, UDP 68. You should note that DHCP uses two different port numbers, one for the client and one for the server. You will need to know both those port numbers if there are any firewalls between the client and the server.

Once the client receives the DHCPOFFER, it will confirm the lease by sending out a DHCPREQUEST. Again, this is sent as a broadcast. It is possible that multiple DHCP servers saw the original request and each responded with a separate lease offer. This broadcast allows all of the servers to see which lease the client is willing to accept, so any extra servers will stop holding a lease for the client. The server with the successful lease will mark the lease as leased. It will also send a DHCPACK which indicates the client now has the lease and is allowed to use those configuration parameters.

Once the client receives the DHCPACK, it writes the particulars of the lease to `/var/db/dhclient.leases` and uses the leased information to participate in a TCP/IP network.

DHCP Time Periods

The client knows it can't use that configuration forever. A lease expiry is clearly marked in the lease. Take a look at the last 3 lines before the `}` in my example lease:

```
renew 3 2003/4/2 00:22:38;  
rebind 6 2003/4/5 02:50:06;  
expire 6 2003/4/5 23:50:06;
```

You'll note there are actually three time periods: renew, rebind, and expire. The syntax for each line is day year/month/day hour:minute:second where 0 means Sunday. In this example, this lease is ready to "renew" on Wednesday April 2, 2003 at 12:22:38 in the morning.

What is the difference between renew, rebind, and expire? Renew is also known as T1 or when the leased time is at 50%. When T1 occurs, the DHCP client will send a DHCPREQUEST to the DHCP server which assigned the lease. Note that the lease itself indicates the address of the DHCP server in this line:

```
option dhcp-server-identifier 2.2.2.21
```

Since the client knows the address of the server, it doesn't have to send a broadcast. If all goes well, the server will receive the DHCPREQUEST and give the client permission to renew the lease. Basically, the client is allowed to reuse the configuration for the original lease time and all three times are bumped up to reflect the new lease period.

Introducing DHCP

Dru Lavigne

If the client doesn't hear back from the DHCP server, it will wait for T2 or when the lease is at 87.5%. This is also known as the rebind period. The client will again send a DHCPREQUEST, but this time it will be a broadcast. Basically, the client is starting to get a bit worried and just wants to get its lease period renewed before it expires. Hopefully, some DHCP server will respond, and again all three time periods will be bumped up to reflect the new lease period.

If things don't go well, the poor DHCP client won't hear back from any DHCP servers. When the expire time occurs, the client is no longer allowed to use its leased configurations and is basically back at square one. The only way to get a lease will be to start from scratch with a DHCPDISCOVER broadcast.

Things work a little bit differently if you reboot your computer before any of the three time periods arrive. At bootup, your FreeBSD system will look for the address of the DHCP server in its /var/db/dhclient.leases file. It will then try to contact the server in order to renew its lease. However, if the DHCP server happens to be unavailable, the client will check to see if it has a non-expired lease. If it does, it will ping the default router from the leased address to see if it still appears to work. If the router responds, it will boot up with the address, and will try contacting the DHCP server in 5 minutes in order to validate the lease.

There are two other possible DHCP messages which you might come across. The first is a DHCPNACK or negative acknowledgment. This message will be sent from a DHCP server if a client requests an address which is no longer valid. This usually occurs when you physically move a computer between subnets.

To prevent that message, you should send a DHCPRELEASE from the client before you physically move a computer to another subnet. To do this on a FreeBSD system:

```
# dhclient -r ed0
```

Typically, you would issue that command just before powering down the system and moving the computer. If you intend to leave the computer up, you won't be getting a new address as -r will also kill the dhclient process. If you wish to renew the lease, simply restart the process:

```
# dhclient ed0
```

Conclusion

This is all probably more than you thought you'd be learning about DHCP for one day. In the next article, I'll concentrate on the DHCP server and the configuration information it needs in order to assign leases to DHCP clients.